REMARKS

Claims 1, 7-9, 13, 15-18, 21, 26-29, 35, 39 and 44 previously were cancelled. Claims 2-6, 10-12, 14, 19, 20, 22-25, 30-34, 36-38, 40-43 and 45-60 are now pending in this application.

Applicants respectfully traverse the rejections without amendment, because Java applets (e.g., Foley) are described in the application as prior art, over which this invention improves.

Claim Rejections under 35 USC 103

Based on Applicants' prior traverse of the combination of Krishna and Kiyono, the Examiner withdrew the prior rejections without responding to the traverse (section 9, page 15). The Examiner introduced an additional reference, Foley et al. (USP 5,706,502).

To put Foley in perspective, we note that Foley motivates and describes her invention as follows:

> an [improved software development and file management system that provides an] Internet-enabled programming environment that allows a user to assemble programs from components that are distributed on diverse Internet nodes or to download entire program folders from another Internet node.

Col. 2, lines 29-33 & 44-46. Foley appears to present an improved feature for what is commonly referred to as an integrated development environment (IDE), without any attention to code versioning, ala CVS. The sections of Foley that refer to applet retrieval and execution, e.g., col. 5, lines 22-27, seem to be what the Examiner most relies upon.

To understand Foley's disclosure of applet retrieval and execution, it is essential to know what an "applet" is. Since this is a Java programming term, Sun is the source of Java, and Foley's patent is assigned to Sun, we refer to the Sun

Java Enterprise System Glossary, http://docs.sun.com/source/816-6873/index.html (accessed May 12, 2004):

> applet (n.) A small application written in the Java™ programming language that runs in a web browser. Typically, applets are called by or embedded in web pages to provide special functionality. A Java applet is a small application program written in the Java™ programming language that can be sent along with a web page to a user's browser. Java applets can perform tasks without having to send a user request back to the server. Instant Messaging client is a Java applet. See also servlet.
>
> servlet (1) (n.) Server-side programs written in the Java™ programming language that web servers run to generate content in response to a client request. Servlets are similar to applets in that they run on the server-side but servlets do not use a user interface.

More information is available from a Basic Java programming guide at:

http://java.sun.com/developer/onlineTraining/Programming/BasicJava1/applet.html

From Sun's materials[1], it appears that the applets to which Foley refers are client-side application programs downloaded and then invoked **by name.**

This application presents Java applets as prior art that creates an opportunity for invention, at pages 2-3. In part:

> A common configuration is a client machine or workstation running browser software that executes applet code (e.g., Java applet) that is downloaded from a server residing on the Internet, such 5 as illustrated in Fig. 1A. Since the protocols used for communication (e.g., TCP/IP) support a variety of platforms, an almost infinite variety of hardware-platform configurations is possible.
>
> A common problem exists in such environments, however. Often, a software development shop or engineering group is required to manage a code base that must exist 10 within the framework of many different customer-specified requirements. Consider, for instance, client-side application software that presents information from a database server (e.g., residing on the Internet) using a browser (e.g., Java-enabled browser) running at a client or desktop computer. For a software development shop that services large telecommunications providers, a substantial amount of customization is often required for 15 ensuring that the deployed application executes properly at the browser and

[1] Applicants do not admit that these Sun materials are prior art; they were accessed after receipt of the Office Action and their original publication status has not been investigated.

> according to custom r specification. For example, the custom r often requires a customized user interface, with customer-specified colors and presentation order. Additionally, customers often want to add new features, so that the application may adapt as technology changes.
>
> Typically, such customization is done using cut and paste techniques (i.e., copying or moving 20 program code from one location to another, or possibly removing it altogether). Regardless of how such changes are effected, once significant changes have been made, the entire program must be rebuilt. As a result, the task of accommodating customer-specific changes to the code is a costly undertaking.

The claimed solution distinguishes over Foley by design, because it improves on the Java applet technology that Foley and Sun describe.

Foley does not provide the claimed dictionary anywhere in the set of JWS Applets 140A (see FIG. 1), as the Examiner asserts. Again, applets are invoked by name, without any intermediate dictionary. One cannot modify a dictionary entry to direct or redirect the run-time handler or run-timer service associated with an applet token, because the applet name directly invokes the applet. An applet invocation, such as:

```
<APPLET CODE=SimpleApplet.class WIDTH=200 HEIGHT=100>
</APPLET>
```

is just another piece of HTML backed up by a client-side program script. Foley does not refer to any dictionary anywhere in his disclosure (he does not use the word dictionary), does not interpose any intermediate dictionary data structure between an applet invoked directly by its name, and does not ever refer to any list of applets. With this perspective on Foley, the Examiner's rejections can readily be addressed.

Regarding independent claim 60, the Examiner relies on Foley to teach in FIG. 1 that JWS Applets 140A contains a list (dictionary) of referenced stored applets 140A1-140A2. There is no support cited in the text of Foley for interpreting FIG. 1 as including a dictionary. Given Sun's public explanations of

how applets work, it is not reasonable to suppose that on of ordinary skill in the art would interpret FIG. 1 as including a dictionary that applies the claimed features to the well-established Java applet technology.

Regarding independent claim 46, the Examiner again (cut and paste from 60) relies on Foley to teach in FIG. 1 that JWS Applets 140A contains a list (dictionary) of referenced stored applets 140A1-140A2. There is no support cited in the text of Foley for interpreting FIG. 1 as including a dictionary. Given Sun's public explanations of how applets work, it is not reasonable to suppose that one of ordinary skill in the art would interpret FIG. 1 as including a dictionary that applies the claimed features to the well-established Java applet technology.

Regarding both of the independent claims 60 and 46, there is no proper motivation to modify Krishna with Foley. Converting Krishna into a development environment would destroy the original purpose of Krishna and would not yield the claimed invention. Development environments always include features that are not part of the production environment and not desirable or practical to implement in production, for security and throughput reasons. It takes more to modify Krishna with Foley than just saying "in order to facilitate the creation or importation of software components such as Java applets (stored locally or on an Internet node)." The purported motivation is unconnected to the considerations of applying Foley to Krishna and unsupported by any suggestion or teaching in either of the references. Moreover, it depends on a misinterpretation of Foley FIG. 1.

Claim Rejections of Dependent Claims under 35 USC 103

The dependent claims are allowable for at least the reasons that the independent claims are allowable.

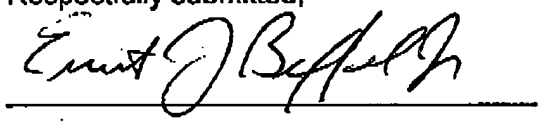
Application No. 09/256,845 **Atty Docket: PUMA 1000-1**

CONCLUSION

Applicants respectfully submit that the claims, as stated herein, are in condition for allowance and solicit acceptance of the claims, in light of these remarks. If the Examiner disagrees and sees amendments that might facilitate allowance of the claims, a call would be appreciated.

Should any questions arise, the undersigned can ordinarily be reached at his office at 650-712-0340 from 8:30 to 5:30 PST, M-F and can be reached at his cell phone 415-902-6112 most other times.

Respectfully submitted,



Dated: 12 May 2004

Ernest J. Beffel, Jr., Reg. No. 43,489

HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
Half Moon Bay, CA 94019
(650) 712-0340 phone
(650) 712-0263 fax